Bayesian Hierarchical Models: Practical Exercises

You will be using WinBUGS 1.4.3 for these practicals.

All the data and other files you will need for the practicals are provided in a zip file, which you can download from
`http://www.bias-project.org.uk/WB2011Man/CourseData/BHMData.zip`.

You should unzip this file and save the contents in the directory `C:\work\bugscourse`**.**

Solutions for all the exercises are also provided in the zip files.

Scripts have also been prepared which can be used to run most of the models as an alternative to using the menu click-and-point interface. You are advised to start by using the menu interface for the first practical, so that you understand the steps involved in setting up and running a model in WinBUGS. However, feel free to use the scripts to run models for the other practicals (or write your own scripts to do this) if you prefer. **If you have copied the data and program files to a directory other than** `C:\work\bugscourse` **you will need to edit all the path names of the files specified in each script.**

Further detailed instructions on using `WinBUGS` **can be found in the handout** *Hints on using WinBUGS*, **and in the on-line WinBUGS User Manual (see the Help menu in** `WinBUGS`**).**

Notes on scripts in WinBUGS 1.4.3

The standard way to control a WinBUGS model run is using the click-and-point menu interface. However, there is also a script language which enables all the menu options to be listed in a text file and run in batch mode. Scripts to run all of the models in the practical exercises are included in the data zip file you were provided with. Using scripts is generally much quicker than clicking-and-pointing. However, we recommend you start with the click-and-point approach so that you understand how WinBUGS works. The click-and-point approach is also better for debugging a model.

The section *Batch mode: Scripts* of the on-line User Manual in WinBUGS gives details of the script language.

To run a script, click on the window containing the script file to 'focus' it, and then select Script from the Model menu.

Notes on scripts:

- The model code, data and each set of initial values called by the script have to be stored in separate files.
- Once the script has finished executing, the analysis is still 'live' and you can continue to use the WinBUGS menus interactively in the usual way.
- The script language currently has very limited error handling, so check that your model compiles correctly and that the data and initial values can all be loaded OK using the usual WinBUGS menu/GUI interface, before setting up a script to carry out a full analysis.

Notes on DIC tool in WinBUGS 1.4.3

- Remember that you should run sufficient burn-in iterations to allow the simulations to converge *before* you set the DIC tool, since you cannot discard burn-in values from the DIC calculations after the monitor has been set.
- There is an FAQ about DIC on the BUGS web site:
  `http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/dicpage.shtml`

Notes on BUGS language

- A list of distributions and their syntax in the BUGS language is given in the on-line Help: `Help: User Manual: Distributions`.
- A list of functions and their syntax in the BUGS language is given in the on-line Help: `Help: User Manual: Model Specification: Logical Nodes`.
- Details of data formats accepted by WinBUGS are given in the on-line Help: `Help: User Manual: Model Specification: formatting of data`.
- Syntax for writing 'loops' and indexing vectors and arrays in the BUGS language is given in the on-line Help: `Help: User Manual: Model Specification: Arrays and indexing` and `Help: User Manual: Model Specification: repeated structures`.

Practical Class 1: Fitting Simple Hierarchical Models using WinBUGS

A. THM example

A simulated set of data (slightly different from that used in the lecture notes) for the THM example can be found in file `thm-dat.odc`. Note that the data are in the 'nested index' format (see lecture 1 slide 23); there are 12 areas for which no data have been collected. The code for fitting a simple non-hierarchical model, assuming independent priors on the mean THM level for each zone, but a common residual error variance across zones, is in file `thm-model.odc`. Two sets of initial values are provided in files `thm-in1.odc` and `thm-in2.odc`.

1. *Non-hierarchical model*

   (a) Run this model in WinBUGS, setting monitors on the zone mean THM concentrations `theta` and on the residual error variance.

   (b) Produce posterior summary statistics for the parameters you have monitored, and box plots of the posterior distribution of the zone mean THM levels. Use the 'special properties' option of the box plot (right click on plot to select 'properties', then select 'special' to display the areas in the box plot in rank order). (See section **Plotting summaries of the posterior distribution** of the hints handout). Comment on the estimates for the areas with no data.

2. *Hierarchical model*

   (a) Edit the model code to change the prior on the zone means (`theta`'s) to a normal random effects distribution with unknown mean and variance, and specify suitable priors for the latter 2 parameters (see lecture 1 slides 12 and 24). Don't forget to also modify your initial values files to include initial values for the additional parameters in your model.

   (b) Include statements in your model code to calculate the variance partition coefficient (VPC) for this hierarchical model (see lecture 1 slide 26).

   (c) Run the model, setting monitors on the zone mean THM concentrations `theta`, the residual error variance, the random effects mean and variance, and the VPC.

   (d) Produce posterior summary statistics for the parameters you have monitored, and box plots of the posterior distribution of the zone mean THM levels. Compare your estimates with those from the non-hierarchical model and comment on any differences.

B. Hierarchical models for binary data: surgical mortality rates

In this question, you will be modelling data on mortality rates following surgery in each of 12 hospitals. The data file `surgical-dat.odc` contains the following variables: `I`, the number of hospitals; `n`, the number of operations carried out in each hospital in a 1 year period; `r`, the number of deaths within 30 days of surgery in each hospital.

The aim of the analysis is to use surgical mortality rates as an indicator of each hospital's performance and to identify whether any hospitals appear to be performing unusually well or poorly.

A suitable model for these data is to assume a binomial likelihood for the number of deaths in each hospital

$$r_i \sim \text{Binomial}(p_i, n_i)$$

where p_i is the mortality rate in hospital i. We must then specify a prior distribution for the p_i's. Here we will treat the mortality rates as exchangeable across hospitals and model them using the following random effects specification

$$\text{logit}p_i \sim \text{Normal}(\mu, \sigma^2)$$

This simply fits a random intercept for each hospital, corresponding to the logit-transformed mortality rate for that hospital. Vague hyperprior distributions are specified for the random effects mean and variance.

The `WinBUGS` code for fitting this model to the surgical data can be found in file `surgical-model.odc`.

1. *Fitting the model:*

   (a) Create two sets of initial values for this model and run the model in WinBUGS. The easiest way to create the initial values is to just specify initial values for the hyperparameters (i.e. the mean and variance (or precision) of the random effects), and once these have been loaded in WinBUGS, use the `gen.inits` option of the Model Specification tool to generate initial values for the random effects (i.e. for the `theta`'s).

   (b) Before you start updating, set *sample* monitors for the vector of parameters `p` (representing the mortality rates for each hospital), `m` and `sigma` (representing the overall mean mortality rate and between-hospital sd in logit mortality rates, respectively) and `QR80` (the 80% quantile ratio — i.e. ratio of mortality odds ratios for hospitals ranked at the 90^{th} and 10^{th} percentiles of the random effects distribution).

   (c) Produce summary statistics and kernel density plots for the posterior samples of all the monitored parameters. Interpret the `QR80`.

   (d) Produce box plots and/or caterpillar plots to compare the mortality rates for each hospital (See section **Plotting summaries of the posterior distribution** of the hints handout). Which hospitals have the 'best' and 'worst' performance in terms of mortality? Are you confident that these hospitals really are the best and worst, respectively?

2. *Posterior probabilities and ranking*

   (a) Edit the model code to include terms in your model code to calculate:

   - the posterior probability that the mortality rate in each hospital exceeds 0.1 (hint: use the `step` function);
   - the posterior probability that the mortality rate in each hospital exceeds the average mortality rate across the 12 hospitals;
   - the rank of the mortality rate for each hospital (hint: use the `rank` function — see lecture 1 slide 33).

   (b) Set monitors on these probabilities and ranks and run the model. You can also monitor the posterior distributions of the rank of each hospital's mortality rate directly by selecting Rank from the Inference menu, and setting a rank monitor for `p`. Note that you should only set this rank monitor *after* you have carried out some burn-in iterations and checked for convergence, as it is not possible to discard iterations from the rank calculations performed by this monitor.

   (c) Obtain summary statistics of the posterior probabilities that the mortality rates exceed the specified thresholds. Produce caterpillar plots or box plots of the posterior distributions of each hospital's rank (you will need to enter the name of the variable that you created in your BUGS model code to represent the ranks in the node box of the graphics tool). If you have set the `WinBUGS` Rank Monitor Tool, use this to now produce summary statistics and posterior histograms of each hospital's rank. Which (if any) of the hospitals can you be confident are better or worse than the others?

   (d) Suppose Hospital 1 in fact had carried out 400 operations and had no deaths. Edit the data file accordingly, and re-run the previous analysis. How confident are you now that hospital 1 is the best (i.e. has the lowest true underlying mortality rate of the 12 hospitals)?

3. *Using the script language*: `WinBUGS` version 1.4 includes a facility for running models in batch mode using a script. The script to run the basic model for the surgical data in `WinBUGS` is in file `surgical-script.odc`. Open this file and look at the commands to make sure you understand them (see section *Batch mode: Scripts* of the on-line User Manual in `WinBUGS` for more details). To run the script, click on the window containing the script file to focus it, and then select Script from the Model menu. (See also the hints on using scripts at the start of the practical exercises sheet). Edit the script to include monitors on the additional variables you have defined in your model code for the posterior probabilities and ranks, and re-run the script.

Practical Class 2: Predictive model criticism and model comparison using WinBUGS: Bristol Royal Infirmary example

The data below are taken from the analysis presented to the Bristol Royal Infirmary Inquiry of data on mortality following paediatric cardiac surgery at Bristol and other specialist centres.

```
list(I=12,
     r=c(25, 24, 23, 25, 42,24,53,26,25,58,41,31),
     n=c(187,323,122,164,405,239,482,195,177,581,143,301))
```

The data are also available in file `bristol-data.odc`.

`I`=12 hospitals were considered in the analysis. The data refer to the total number (`n`) of open heart surgery operations carried out on children under 1 year old by each centre between April 1991 and March 1995, and the number of deaths (`r`) within 30 days of the operation.

One of the aims of the analysis is to explore whether an assumption of exchangeability is reasonable for the mortality rates in all 12 hospitals, or whether there is any evidence that the mortality rates for one or more of the hospitals do not appear to be drawn from the same (parent) distribution as the rest. Additionally, the literature shows a consistent relationship between volume of surgery (i.e. the number of operations carried out) and the outcome for a range of operations and so it may be appropriate to consider whether volume of surgery (i.e. `n`) should be included as a covariate in this analysis.

1. *Predictive model criticism:* The basic model for the Bristol data is similar to the model for the surgical data that you fitted in practical 1 (i.e. a logistic random effects model). Using the `surgical-model.odc` code as a starting point, modify it to fit the Bristol data. Include statements in your model code to calculate posterior and mixed predictive p-values to help identify whether any of the hospitals appear to have unusual mortality rates (see lecture 3 slide 38).

   (a) Run this model and examine the predictive p-values. Which hospital(s) appear to have an unusual mortality rate?

   (b) Produce kernel density plots of the predicted number of deaths in each hospital (for both posterior and mixed predictions), and visually compare these to the observed number of deaths in each hospital. The corresponding p-values summarise how far in the tails of the predictive distribution each observation lies.

2. *Model comparison:* Define a set of plausible candidate models for these data. A binomial likelihood seems reasonable for these data, so focus attention on comparing different models for the hospital-specific mortality rates. Some potential models to consider include:

(1)	$p_i \sim \text{Uniform}(0, 1)$	independent effects
(2)	logit $p_i \sim \text{Normal}(\mu, \tau)$	all hospitals exchangeable with Normal prior
(3)	logit $p_i \sim \text{Student-t}(\mu, \tau, 4)$	all hospitals exchangeable with t_4 prior
(4)	$\begin{cases} \text{logit } p_i \sim \text{Normal}(\mu, \tau) \\ \qquad\qquad i = 1, .., 10, 12 \\ \text{logit } p_{11} \sim \text{Normal}(0, 0.000001) \end{cases}$	$\begin{cases} \text{hospitals 1–10, 12} \\ \text{exchangeable with Normal prior;} \\ \text{independent effect for} \\ \text{hospital 11 (Bristol)} \end{cases}$
(5) − (7)	logit $p_i = q_i + \beta * n_i$ $q_i \sim$ priors as for logitp_i in (2)-(4)	covariate models

(a) Fit each model in turn and calculate the DIC (see hints on using DIC at the start of the practical exercises sheet). You may find it easier to set up a script to run the models, and just change the name of the model file and initial values file for each new run. See `bristol-script.odc` for example. Which model is best supported according to DIC?

(b) For models (5)-(7), monitor `beta`, the effect of volume of surgery on outcome. Produce summary statistics for β or $\exp(\beta)$, where for the latter, you need to add a line in the WinBUGS code to transform β to $\exp(\beta)$. Comment on the results. Should the volume of surgery be included as a covariate in the analysis?

(c) Compare the posterior distributions of the hospital mortality rates under each model, and interpret the differences. (Hint: the comparison is easiest to do by producing box plots of the mortality rates `p` under each model).

(d) (*optional if you have time) For the best supported model, include statements to calculate both posterior and mixed predictive p-values (as in part 1). Do the p-values for some hospitals remain extreme, or does this model better explain the unusual mortality rates? (Note: you will need to think about whether the mixed p-values really make sense for this model).

If you get stuck, have a look at the solutions file!

Practical Class 3: Sensitivity to priors

This question uses the data from the Surgical example in Practical class 1, and investigates sensitivity to priors on the random effects variance.

1. Modify the model code in `surgical-model.odc` to specify each of the following priors in turn for the variance of the Normal distribution assumed for random effects:

   (1) Gamma(0.001, 0.001) on the random effects precision (this is the prior you have already used)

   (2) Uniform(0, 100) on the random effects variance

   (3) Uniform(0, 100) on the random effects standard deviation

   (4) An informative prior that reflects the belief that you think it unlikely that the odds of death following cardiac surgery vary by more than about 40-50% between hospitals. From the table given in the lecture notes (slides 15, 16 lecture 2), we see that a random effects sd = 0.1 represents around 50% variation in odds between 90% of hospitals, so you could specify a Uniform(0, 0.1) prior on the random effects standard deviation. However, this is very restrictive, and allows no possibility for greater variability between hospitals. A better choice may be to specify a half-normal prior distribution with lower bound at zero, and 95^{th} (say) percentile at 0.1. Since the tail area probabilities of a half normal are double those of an untruncated normal, we need to find the variance for an untruncated normal that has 0.1 as its 97.5^{th} percentile. An N(0, 0.05^2) distribution has these properties (approximately), so in WinBUGS, specify the prior as

   ```
   sigma ~ dnorm(0, 400)I(0, )
   ```

   where `sigma` (say) is the name of your random effects standard deviation, the `I(0, )` notation is used to specify the lower bound, and a value of 400 for the precision equals $1/0.05^2$.

   Remember that you need to specify initial values for the stochastic parameters (i.e. the parameters with priors) in your model, so if the prior is specified on the random effects precision, you need an initial value for the random effects precision, but if the prior is on the random effects standard deviation, the initial value must be for the standard deviation not the precision, etc.

2. For each prior in turn, run the model and produce summary statistics and kernel density plots of the posterior distribution of the random effects standard deviation, the 80% quantile ratio (`QR80`), and each of the hospital mortality rates. Compare these across priors. Is there strong evidence of sensitivity to the first 3 priors (which are all intended to represent vague prior information)? How are your estimates affected by the informative prior?

Practical Class 4: Modelling longitudinal data using `WinBUGS`

A. Repeated measurements of CD4 counts in HIV-infected patients

This example uses (simulated) data from a clinical trial comparing two alternative treatments for HIV-infected individuals. 80 patients with HIV infection were randomly assigned to one of 2 treatment groups (`drug` = 0 (didanosine, ddI) and `drug` = 1 (zalcitabine, ddC)). CD4 counts were recorded at study entry (time `t` = 0) and again at 2, 6 and 12 months. An indicator of whether the patient had already been diagnosed with AIDS at study entry was also recorded (`AIDS` = 1 if patient diagnosed with AIDS, and 0 otherwise). The data can be found in files `CD4-dat.odc` and `CD4-time-dat.odc`.

1. *Fitting a non-hierarchical linear model*

   The file `CD4-model.odc` contains code to fit a non-hierarchical (i.e. fixed effects) linear regression model to examine the effect of drug treatment, AIDS diagnosis at study entry, and time since entry to clinical trial on CD4 count (the model is similar to the non-hierarchical model for the HAMD example discussed on slides 8, 9 and 12 of lecture 4). Two sets of initial values can be found in `CD4-inits1.odc` and `CD4-inits2.odc`.

   (a) Run the model and monitor the slope and intercept parameters, the regression coefficients for the effects of treatment and AIDS, and the residual error variance. You should also set a monitor on the matrix of parameters corresponding to the 'fitted values' (i.e. the mean of the normal likelihood you have specified for the CD4 counts) for patients 1 to 10 (i.e. monitor `mu[1:10, ]`). You will need the samples of these fitted values to produce the model fit plots (see below). After convergence, monitor the DIC as well.

   (b) Produce summary statistics of the monitored variables, and note the DIC.

   (c) Produce 'model fit' plots to show the 2.5%, 50% and 97.5% quantiles of the fitted regression line for each of patients 1 to 10 (select 'model fit' from the 'compare' tool on the 'Inference' menu).

2. *Fitting a hierarchical linear model*

   Modify your code for the previous model to include a random intercept and a random slope (i.e. time coefficient) for each patient. Treat the coefficients for the effects of drug treatment and AIDS as fixed (i.e. not random effects) as before. Assume independent prior distributions for the random intercepts, and for the random slopes. Remember that you will also need to give hyperprior distributions for the parameters of these random effects distributions (i.e. for the random effects means and variances) and to specify initial values for these hyperparameters (you will then also need to use `gen inits` to generate initial values for the random effects themselves).

   (a) Run the model and monitor on the same parameters as before, plus the patient-specific slope and intercept parameters (random effects) for patients 1 to 10 (don't

monitor all 80 patients as storing the sampled values for all the random effects will quickly start to 'clog-up' the computer's memory), and the standard deviations of the random effects. Monitor the DIC as well.

(b) Produce summary statistics and plots as before, and compare your results with those from the non-hierarchical model. Compare the DIC values to choose the most appropriate model.

3. *Fitting an AR1 model (*optional if you have time)*

Modify your code for the non-hierarchical linear regression model (part 1) to explicitly model the autocorrelation between the recordings of CD4 counts for each patient. You should set up an autoregressive model of order 1, AR(1), assuming that the residuals are serially correlated, as described for the HAMD example in Lecture 4 (see slide 25 for WinBUGS code). You will also need to specify initial values for the autoregressive coefficient that will be included in the new model.

(a) Run the model and monitor the same parameters as in part 1, plus the autoregressive coefficient. Also, monitor the DIC.

(b) Produce summary statistics as before, and compare your results with those from the non-hierarchical and random effects models. Using the DIC values, compare the fit of the AR(1) model with the other two models.

B. Cognitive decline: The Ageing example

In this practical, you will investigate two alternative models for the ageing study discussed in lecture 4: (a) specifying independent (rather than bivariate) distributions for the subject-specific random intercepts and random slopes and (b) fitting the hierarchically non-centred version of the model. In this practical, we use only 40 subjects, sampled randomly from each education stratum (edu=0 and edu=1); each subject has 4 MMSE scores, i.e., the complete case scenario. The sampled data can be found in `ageing-data.odc`. We do not include ceiling effect here but for those who are interested, the codes are in `ageing-randcoefEduCeiling.odc`.

The WinBUGS code for fitting the hierarchically centred model with education effects is given in `ageing-randcoefEdu.odc`. You are advised to make a copy of this file, as it will be used as a template for the following two models.

1. *Fitting a model with independent distributions on the random effects*

(a) Modify the file `ageing-randcoefEdu.odc` to specify independent Normal prior distribution on the random effects. That is, $\alpha_i \sim \mathrm{N}(\mu_{\alpha,i}, \sigma^2_\alpha)$ and $\beta_i \sim \mathrm{N}(\mu_{\beta,i}, \sigma^2_\beta)$ where $\mu_{\alpha,i} = \eta_0 + \eta_1 \cdot edu_i$ and $\mu_{\beta,i} = \gamma_0 + \gamma_1 \cdot edu_i$. You should assign the following weakly informative priors for the random effects standard deviations: $\sigma_\alpha \sim \mathrm{Uniform}(0, 30)$ and $\sigma_\beta \sim \mathrm{Uniform}(0, 10)$. In the original code, you need to remove the parts that relate to the joint modelling of α_i and β_i.

(b) Modify the initial values files provided (in files `ageing-ind-inits1.odc` and `ageing-ind-inits2.odc`) to include initial values for the random effects standard deviations that you have just specified in the model code. (Note, initial values for the random effects themselves can be generated directly in WinBUGS after the initial values files have been loaded).

(c) Check model, load data and the initial values and use `gen inits` to generate initial values for the remaining parameters. Set monitor on `eta0`, `eta1`, `gamma0`, `gamma1`, `mmse.high.edu` and `rate.high.edu`. Also monitor the random effect terms (`alpha` and `beta`), all variances, pD and DIC (only set this after the burn-in).

(d) Set at least 5000 iterations for burn-in, and another 10000 iterations for computing posterior summary.

(e) Complete Table 1. Compare and comment on the differences/similarities between the models with independent and joint random effects priors.

(f) (*optional if you have time) It is always recommended to perform sensitivity analyses to examine how robust the conclusions are to different priors. Refit the model with independent distributions on the random effects with the following two sets of hyperpriors for the random effects variation:

i. $\sigma_\alpha \sim \text{Uniform}(0, 100)$ and $\sigma_\beta \sim \text{Uniform}(0, 100)$;

ii. $\sigma_\alpha^{-2} \sim \text{Gamma}(0.001, 0.001)$ and $\sigma_\beta^{-2} \sim \text{Gamma}(0.001, 0.001)$.

2. *Fitting the hierarchically non-centred model with joint modelling of the random effects and education*

(a) Modify the file `ageing-randcoefEdu.odc` to fit the non-centred model on slide 4-43;

(b) Complete Table 2 and comment on the differences/similarities (e.g., on convergence and parameter estimates) between the centred and the non-centred parameterisations. Specifically, comment on the differences between the `alpha` and `beta` estimates. Why do they differ?

Table 1. Posterior means (and 95% credible intervals) for the parameter estimates and DIC from the hierarchically centred models with (a) a MVN joint prior on α_i and β_i and (b) independent distributions for α_i and β_i with $\sigma_\alpha \sim \text{Uniform}(0, 30)$ and $\sigma_\beta \sim \text{Uniform}(0, 10)$ hyperpriors.

Parameters	Description	Joint prior/Centred	Independent priors
η_0	MMSE at age 75 (edu=0)	28.71 (28.09, 29.35)	
η_1	Effect of education on MMSE at 75	0.15 (-1.13, 1.42)	
γ_0	Cognitive rate of change (edu=0)	-0.13 (-0.28, 0.01)	
γ_1	Effect of education on rate of change	0.08 (-0.20, 0.36)	
Σ_{11} (or σ^2_α)	Variance of alpha	1.75 (0.93, 3.14)	
Σ_{12}	Covariance of alpha and beta	-0.13 (-0.36, 0.04)	
Σ_{22} (or σ^2_β)	Variance of beta	0.13 (0.08, 0.21)	
`cor.alpha.beta`	Correlation between alpha and beta	-0.26 (-0.57, 0.10)	
α_1	Alpha for subject 1	28.47 (27.03, 29.93)	
α_2	Alpha for subject 2	28.97 (26.85, 31.15)	
β_1	Beta for subject 1	-0.05 (-0.35, 0.24)	
β_2	Beta for subject 2	-0.06 (-0.27, 0.16)	
σ^2	Residual variance	2.34 (1.77, 3.09)	
Dbar		588	
pD		59	
DIC		648	

Table 2. Posterior means (and 95% credible intervals) for the parameter estimates and DIC from the hierarchically centred/non-centred models.

Parameters	Description	Joint prior/Centred	Non-centred
η_0	MMSE at age 75 (edu=0)	28.71 (28.09, 29.35)	
η_1	Effect of education on MMSE at 75	0.15 (-1.13, 1.42)	
γ_0	Cognitive rate of change (edu=0)	-0.13 (-0.28, 0.01)	
γ_1	Effect of education on rate of change	0.08 (-0.20, 0.36)	
Σ_{11} (or σ^2_α)	Variance of alpha	1.75 (0.93, 3.14)	
Σ_{12}	Covariance of alpha and beta	-0.13 (-0.36, 0.04)	
Σ_{22} (or σ^2_β)	Variance of beta	0.13 (0.08, 0.21)	
`cor.alpha.beta`	Correlation between alpha and beta	-0.26 (-0.57, 0.10)	
α_1	Alpha for subject 1	28.47 (27.03, 29.93)	
α_2	Alpha for subject 2	28.97 (26.85, 31.15)	
β_1	Beta for subject 1	-0.05 (-0.35, 0.24)	
β_2	Beta for subject 2	-0.06 (-0.27, 0.16)	
σ^2	Residual variance	2.34 (1.77, 3.09)	
Dbar		588	
pD		59	
DIC		648	

Practical Class 5: Further hierarchical modelling using `WinBUGS`

Hierarchical variances: Analysis of N-of-1 trials

The data for the N-of-1 example from lecture 5 is in file `nof1-data.odc`. The file `nof1-model.odc` contains basic WinBUGS code to fit a hierarchical model to these data allowing hierarchical priors for both the subject-specific mean treatment effect and the subject-specific (log) variances.

1. Include statements in this model code to calculate the probability that the active drug is superior for each patient (i.e. that the treatment effect is positive), together with the overall probability that the active drug is superior. Also include statements in your model code to calculate the empirical mean and variance of each subject's data (see the solutions if you are stuck).

   (a) Run this model and produce caterpillar plots of the mean treatment effect for each subject, `theta` and also the within-subject variance (`sigma2`).

   (b) View the values of the empirical mean and variance of each subject's data using Info → Node info → values. Compare the raw estimates of the means and variances for subjects 1, 9 and 23 with the estimates from the hierarchical model. Can you explain the differences?

   (c) What is the probability that the active treatment is superior for each patient? What is the overall probability that the active treatment is superior?

2. Edit the code to include log dose as a covariate in the model for the mean treatment effect for each subject. Remember to also edit the initial values file to include initial values for any new parameters you introduce. Also remember that centering any covariates about their mean is always a good idea.

   (a) Run this model.

   (b) What is the posterior estimate of the effect of dose on response to treatment?

   (c) Compare your results with those from the model without dose. Is it important to adjust for dose in this analysis?

Practical Class 6: Handling missing data using `WinBUGS`

A. Missing covariate data: Low birth weight example

This question is based on the low birth weight example discussed in Lecture 6 (see slides 31-36). We will be using a simulated set of data, with a smaller number of records (500), magnified effects and a substantially higher proportion of low birth weight. The full data can be found in `lbw-full-data.odc`. Here `lbw` is a binary indicator of whether the infant has full term low birth weight, `thm` is a binary indicator of whether trihalomethane concentrations are high (1 = low; 2 = high), `age` is a 4-level categorical variable denoting the mother's age group (1 = $\leq$25 yrs; 2 = 25-29 yrs; 3 = 30-34 yrs; 4 = $\geq$35 yrs), `sex` is a binary indictor of baby gender (1 = female; 2 = male), `dep` is a deprivation index (centered) and `smoke` is a binary indictor of maternal smoking during pregnancy.

We artificially replace some of the values of the `smoke` variable with missing values, and this version of the data can be found in `lbw-missing-data.odc`.

The code to fit a (non-hierarchical) logistic regression model to the full data is given in the file `lbw-model.odc`. Two sets of initial values can be found in `lbw-inits1.odc` and `lbw-inits2.odc`. The models for this practical do not run very quickly, so just run 1000 iterations for burn-in and a further 1000 iterations for computing the posterior summary.

1. Fit the logistic regression model to the full data (`lbw-full-data.odc`) and produce summary statistics of the odds ratios. You will use these to assess the performance of the models fitted to the missing data.

2. Fit the logistic regression model and produce summary statistics using the complete cases (after the missingness is imposed). The complete cases data are given in file `lbw-cc-data.odc`. Use the same model `lbw-model.odc` and initial value files as in part 1, but now the number of records is reduced from 500 to 198. How are the odds ratios affected (in particular, look at the ORs of lbw associated with `thm`, `dep` and `smoke`)?

3. Edit the model in `lbw-model.odc` to fit a logistic regression model to the data with missing values for the `smoke` covariate. You should assume that the missing data mechanism is ignorable. This means that you just need to specify a model to impute the missing `smoke` covariate, but do not need to worry about explicitly modelling the missing data mechanism. For example, model the missing smoke data by specifying a distribution for `smoke`, i.e. using

   $$\texttt{smoke}[i] \sim dbern(q),$$

   and a vague prior for q, the probability that a mother smoked during pregnancy, e.g. $q \sim dbeta(1, 1)$. (If you prefer, just use the code already provided for this model in file `lbw-missing-model.odc`). Two sets of initial values can be found in the files `lbw-missing-inits1.odc` and `lbw-missing-inits2.odc`. The initial values provided are only for the parameters in the models, so you will need to use gen inits to generate initial values for the missing values of `smoke`.

   (a) Run this model in WinBUGS and monitor and obtain summary statistics for the odds ratios, the posterior distribution of q and a subset of the imputed values for

`smoke`, say for individuals 1–10 (only the individuals with missing `smoke` will be monitored). Note that, out of the mothers with observed values of `smoke`, one third smoked during pregnancy.

(b) Compare the odds ratios of interest with those from the full data analysis and the complete case analysis. Is this a good imputation model?

(c) Why do the posterior distributions of the imputed values for `smoke` vary across individuals?

4. Edit the model code to fit a second model to the missing data, including all the regressors in the analysis (outcome) model as predictors in the imputation model for `smoke` (see lecture 6 slide 34). Include vague prior distributions for all the parameters in the imputation model, e.g. phi0 $\sim dnorm(0, 0.0000001)$. You will also need to edit the initial value files. Run this model and compare your estimates with those from the previous models. Which covariate imputation model do you prefer?

B. Missing response data: Income example

This example uses a small subset of sweep 1 data taken from the Millennium Cohort Study (MCS), relating to 200 single main respondents (usually mothers) who are in work and not self-employed. The motivating question concerns how much extra an individual earns if they have a higher level of education. The full data can be found in `income-full-data.odc`. `HPAY` is a continuous variable containing the log of hourly pay (the distribution of hourly pay is skewed, so we use a log transform to achieve approximate normality). `STRATUM` is a 9-level categorical variable, required to take account of the structure in the data as MCS is stratified by UK country (England, Wales, Scotland and Northern Ireland), with England further stratified into three strata (ethnic minority, disadvantaged and advantaged) and the other three countries into two strata (disadvantaged and advantaged). `AGE` is a continuous variable denoting the repondent's age. `REG` is a binary indicator for region (1 = London; 2 = other). `EDU` is a 3-level categorical variable indicating the level of National Vocational Qualification (NVQ) equivalence such that 1 = none or NVQ 1; 2 = NVQ 2/3 (0/A-levels) and 3 = NVQ 4/5 (degree).

The code to fit a regression model to the full data is given in file `income-model.odc`, and two sets of initial values can be found in `income-inits1.odc` and `income-inits2.odc`. Note that we assume t_4 errors for robustness to outliers.

We artificially replace some of the values of the `HPAY` variable with missing values, and this version of the data can be found in `income-missing-data.odc`.

1. Fit the regression model and produce summary statistics using the full data. Is income level affected by education level?

2. Fit the regression model and produce summary statistics using the incomplete data. Whereas for missing covariates you must add a sub-model to take account of missingness, you do not need to do this for missing responses, so there is no need to edit `income-model.odc`. However, because you are not adding a model of missingness for

the response, you are making a strong assumption that the missingness is 'ignorable'. Do your conclusions about the effect of higher education levels change?

3. Now extend the code to add a selection model for the missingness, which corresponds to assuming that the missingness mechanism in not ignorable. You will need to include a binary indicator for missing pay, `PAYID`, (0 = observed, 1 = missing) in your data, so now use file `income-missing-data2.odc`. Just use `HPAY` as the regressor in the logistic equation. The following weakly informative priors are recommended for the coefficients in the missingness model: $\theta \sim dlogis(0, 1)$ and $\delta \sim dnorm(0, 1.48)$, where θ is the constant parameter and δ the parameter associated with `HPAY`. These priors incorporate a 95% prior belief that the change in the odds of being missing is between 1/5 and 5 for a one unit change in `HPAY`, which we believe should not be too restrictive. You will also need to amend the initial value files, for example `theta=0, delta=-0.5` and `theta=0, delta=0.5`. What does this model suggest about education level? Do you think low or high hourly pay rates are more likely to be missing?

4. Estimation of the δ parameter is known to be difficult, and is dependent on the validity of the distributional assumptions in the model of interest. We cannot be sure that these assumptions are correct, so sensitivity analysis is essential. What happens if you assume Normal rather than t_4 errors? What happens if you fix δ to different values, for example $\delta = 1$?